ソフトウェア基礎論 20－3
（1987. 2. 13）

プログラムテストに用いるパスジェネレータ作成に関する研究

柳沢 隆夫
芝浦工大

本論文は、プログラムの自動的なテストパス発生に於いて生じる、2つの問題のためのアルゴリズムを考慮している。
これらの問題は、有向グラフの全ての辺を含む最少のパス集合を決定することと、有向グラフの指定された辺の集合を通る最少のパス集合を決定することである。

On Problem in the Path Generater for Program Testing

TAKAO Yanagisawa
SHIBAURA Institute of Technology
TOKYO JAPAN

In this paper we consider algorithm for tow problems that arise in automatic test path generation for program: the problem of determining a minimal set of paths which contain all the edges of a directed graph and the problem of determining a minimal set of paths which is through a specified edges of a directed graph.

1. はじめに

プログラムテストのための有向サイクルを含まない有向グラフの各々の辺を含む最少パス集合を求める研究は、Deb[1]によって、割当アルゴリズムを用いてグラフをチェインに分割し、輸送アルゴリズムによって、チェインをパス集合に拡張して行われた。その後 Ntafos[2]によって、最少流量法を用いて求められている。

一方、有向サイクルが含まれていることを許したグラフに対する、最少パス集合の導出についての研究も、これまでに若干行われている[3][4]。

柳沢は先に、有向サイクルの有無に拘らず、グラフの指定された頂点とパスを通る最少パス集合を求める方法（有向グラフの各節を含む最少パス集合に帰着させて解く）を報告した。しかし そこでは 有向サイクルのコンデンスと強連結成分内のサブパスの導出に、D. F. S法(depth-first search)が用いられている。

本研究は、前回における有向サイクルのコンデンスや、コンデンスされたものを元に戻す手続きが省略されれば、さらに効率的であることや、最短のパス集合を求めるとさらに有効性が増すという問題を、前回とは異った接近を試み、各有向辺を含み、最少かつ最短なパス集合を求める解法を行った。

この解法は、最少で最短なパス集合を導出する問題とネットワーク理論による最少コストフローのS－T単位順枝列フロー分解とが等価であることを示し、続いて、大きなグラフの最少コストフローの導出とS－T単位順枝列フローへの分解に、簡単で効率的なアルゴリズムが応用出来ることを示すことにより行われる。

ところで、プログラムテスト法として、先ずプログラムグラフのテストパスを決定し、次にこれを通る入力データを算出して行うものは、テストパスが不実行パスであったときは破局となる。もしプログラムテストパスがプログラムのブランチカバーの最少のものであったときは、この問題は重大となる。この問題に対し、Kundu[5]は全てのブランチをカバーするテスト法として、未だテストされていないパス・プレディケイト(predicate、そのパスを実行する入力変数の満足すべき範囲)より、次のテストデータを決定し、この方法を全てのプログラムブランチがカバーされるまで続けるものを提案したが、この方法も、S－Tパス集合が大のときは難解なものとなる。

本研究はテスト法として、先ずプログラムのスタートとディシジョンの直ぐ後に、カウンターを挿入し、いくつかの入力データをランダムに入力してテストを行い、次に残りの未だ未テストの部分にテスターを振り向けるテスト法の問題を扱う。

この問題は、プログラムグラフの指定された辺を通るパス集合を求めるものと等価である。本研究は、未テスト辺を含む最少で最短なテストパスをグラフ理論を応用して導出するとき生じた種々の問題を検討した。

2、本研究で使用される主な用語

有向グラフG：空でない集合VとVと素な集合Eと、写像ψ（：$E \to V \times V$）で構成され、Vの元をGの頂点，Eの元を有向辺という。$G=[V, E, \psi]$であらわす。

ネットワークN：有向グラフGと区間重み関数Iと流動f（フロー）とから構成される。Gは節点集合Vと付向枝集合Eより構成される。Iは枝集合上で定義され，fの上，下限を与えるところの閉区間$I(e)$を与える。fは枝集合上で定義され，フロー公理

$$\sum_{e \in \omega^{+}(x)} f(e) = \sum_{e \in \omega^{-}(x)} f(e) \quad (\forall x \in V)$$

を満足する。ここで，$\omega^{+}(x)$は節点xより出る枝集合，$\omega^{-}(x)$は節点xへ入る枝集合を示す。$N=[G, I, f]$で表わす。

許容フロー：各枝を流れるフローが，枝の閉区間に入っているフロー。

フローコスト：単位当りの枝流量を維持するのに必要なコスト率が，各枝に定数重み$r(e)$として与えられているときのフローfの総コスト

$$\left(\sum_{e \in E} r(e) \cdot |f(e)|\right)$$

隣接する：有向グラフGに於いて，$\psi(e_m)=(v_i, v_j)$ $v_i, v_j \in V$なるe_mが存在するならば，v_iはv_jに隣接しているという。

到達する≦：Gに於いて，全ての$i=1\,2\,3\cdots m-1$に対して，v_iがv_{i+1}に隣接している頂点の連続$v_1 v_2 v_3 \cdots v_m$が存在するならばv_1はv_mに到達可能という。

パスJ（順枝列）：各節が次の節に隣接している節の連続（各付向枝の始節が次の付向枝の終節になるような付向枝の連続。並列枝の無いネットワークでは，節の連続で表わしてさしつかえない）

有向サイクルL（順閉路）：始めと終りの節点が同一の順道

単位順枝列フローf_J：順枝列に含まれる付向枝では$f(e)=1$，それ以外では$f(e)=0$

フローマトリックスF

$$f_{ij} \begin{cases} f(e_m) & v_i \text{が} v_j \text{に隣接しており、} \psi(e_m)=(v_i, v_j) \text{のとき。} \\ 0 & \text{それ以外} \end{cases}$$

強連結成分：どの$v_i, v_j \in V$に対して，$v_i \leqq v_j$，$v_j \leqq v_i$かつGの極大な部分グラフ。

3．最少で最短なS－Tパス集合の導出（最少フロー法）

この節は、有向グラフGの全ての有向辺を含む最少で最短なS－Tパスを導出する方法を述べる。

この方法は、グラフの各辺にフローの下限として1，コスト率として1を与えそしてこのグラフの最少コストフローを構成し、そしてそれから，フローマトリックスを単位順道フローと単位順閉路フローに分解，結合して，最少で最短な

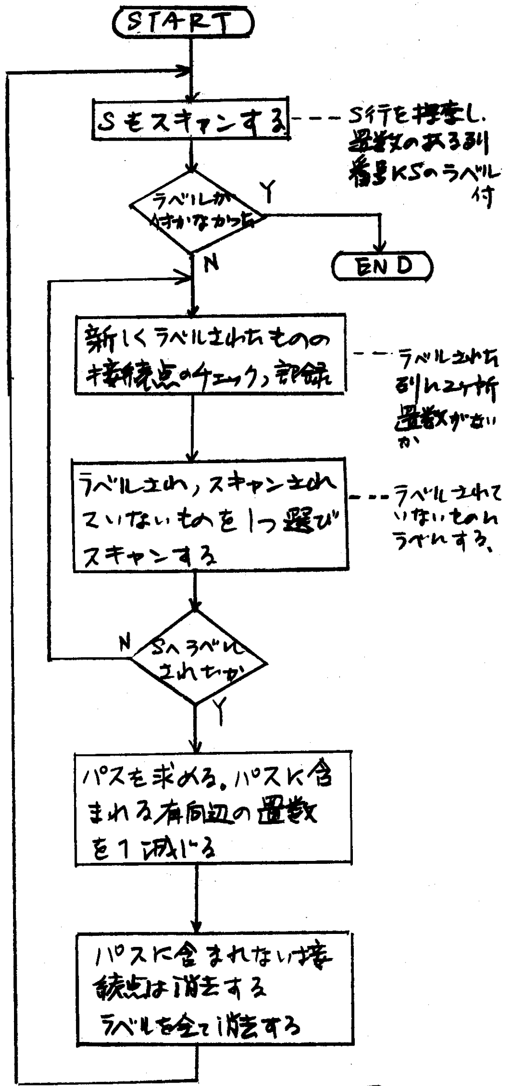
S－Tパスを求める。

次に上記の方法で最少で最短なS－Tパスが求められることを述べる。

(1)、任意の最少フローは、S－T単位順枝列フローに分解出来る。

(2)、最少フローのS－T単位順枝列フロー分解に対応するS－Tパス集合は、Gの全ての有向辺を含む最少のS－Tパス集合である。

(3)、最少コストフローのS－T単位順枝列フロー分解に対応するS－Tパス集合はGの全ての有向辺を含む最少で最短のS－Tパス集合である。



図－1

3．1　最少コストフローの構成

1、許容フローを構成する

2、SよりTへ最少フローを流す。

3、最少コストフローをフローマトリックスに構成する。

3．2　フローマトリックスよりパス集合の導出

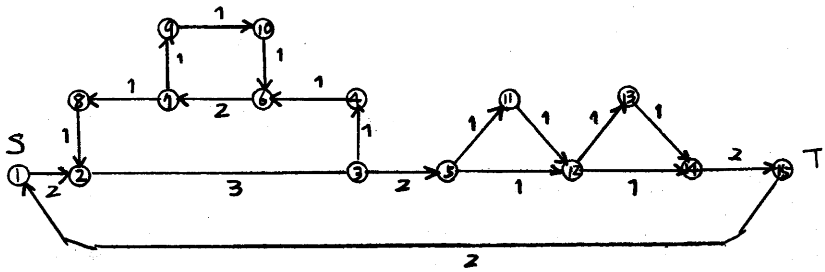
フローマトリックスから，最少で最短なS－Tパス集合の導出は次のように行う。

上記により導出された，最少コストフロー・マトリックスに図－1の手続きを行う。

図－1の手続の終了時点で、未だ出入フローのある全ての接続点をSとし、行と列をかえて，図－1の手続を同様に行う、

上記の手続が終了した時点で未だ出入フローのある接続点があったら、さらに同様に上記手続を行う。

これらの手続を繰返し、接続点が全く生じなかった段階で終了する。



図－2

図−２は最少コストフローが構成されたネットワークの例を表わしている。付向枝に付けられた値は、フロー値を表わしている。S−Tパス集合の導出は、才１段階で例えば，（１２３５１１１２１４１５），（１２３５１２１３１４１５），（１２３５１２１３１４１５）のパスが導出される。接続点は２つの有向枝が入っている節のため、この場合はA点となる。才２段階で，（２８７６４３）が導出され、逆向に変換されて、上記のパスのいずれかに連結される。例えば，（１２３４６７８２３５１１１２１４１５）というパスが構成される。接続点はBとなる。才３段階で、（１２３４６７９１０６７８２３５１１１２１４１５），（１２３５１２１３１４１５）の最少のS−Tパスが得られる。

4、未テスト辺を含む最少で最短なS−Tパス集合の導出（最少フロー法）

(1)、グラフに有向サイクルが含まれない場合

図−３の手続で、未テスト辺を含む最少で最短なS−Tパス集合が求められる

未テスト辺にフローの下限１を与え、それ以外の辺に０を与える。
許容フローを導出する

↓

各辺にコスト１を重みとして与え、最少コストフローを導出

↓

図−１の手続きを行う

図−３

(2)、有向サイクルが含まれるのを許したグラフの場合

Sより未テスト辺に到るパス集合並に未テスト辺よりTに到るパス集合を求め、各々にフロー１を流す

↓

未テスト辺に下限１，それ以外に下限０を与える。グラフの各々の辺にコスト１を割当て、最少コストフローを導出する。

↓

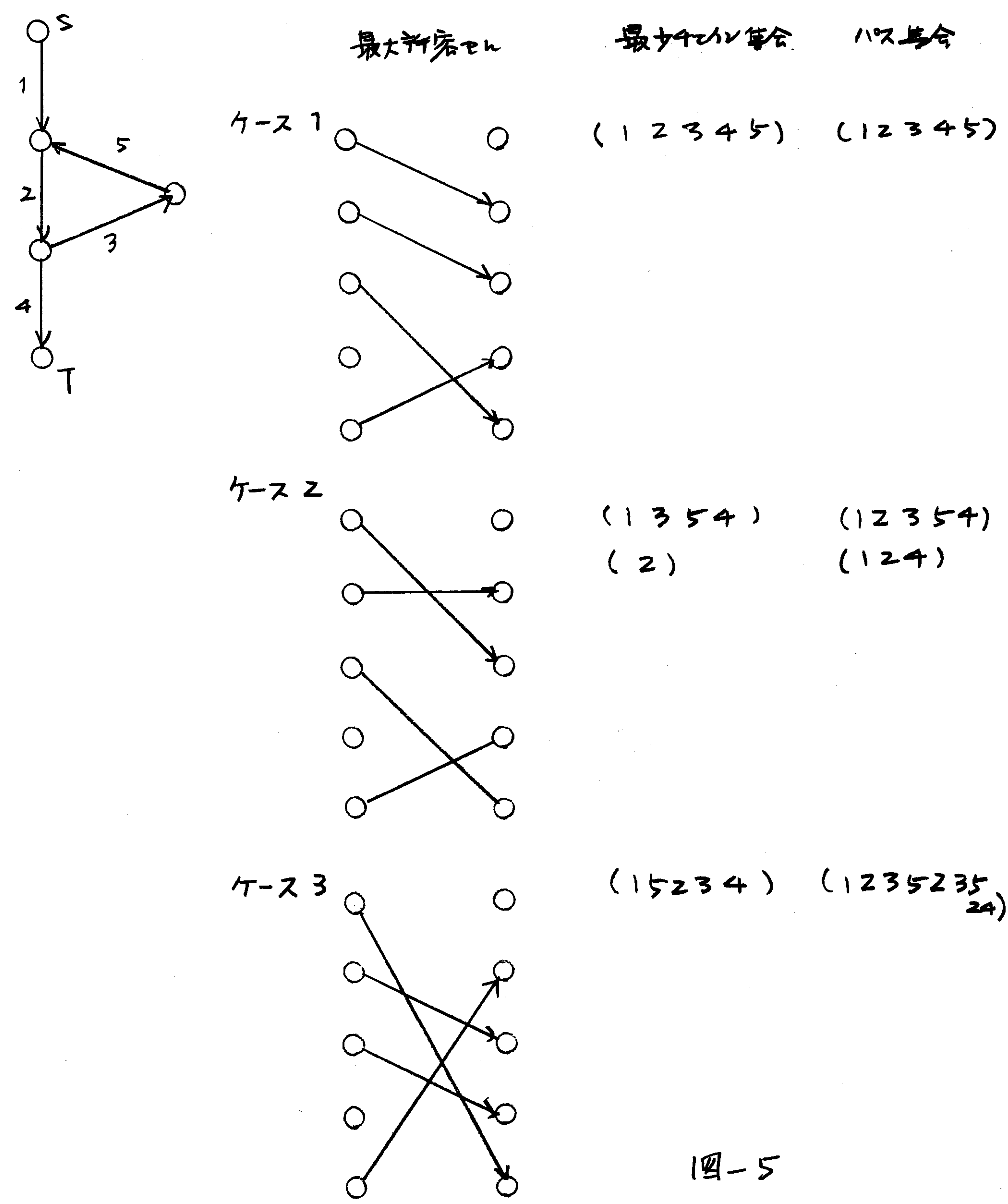
図−１の手続きを行う

図−４

5、最少で最短なS－Tパスの導出法（最大許容セル法）。

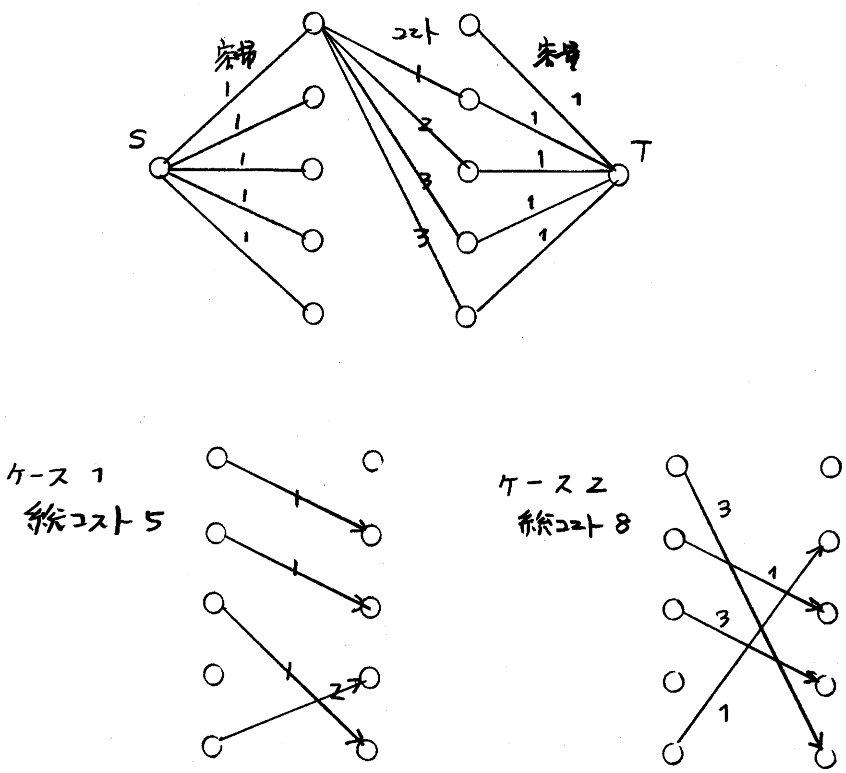
（許容セル：$b_{ij}=1$となる到達可能行列Bのセル（i,j）、独立な許容セル：任意の2つの許容セルがBの同じ列、あるいは同じ行に属さないならばその許容セルの集合は独立という。最大許容セル：最大な独立な許容セル）

有向サイクルのあるグラフに最大許容セル導出法を適用して、最少パス集合の導出を行うと、例えば 図−5のようになり、最少チェイン集合は1と2と3のケースが任意に選ばれて求められる。



図−5

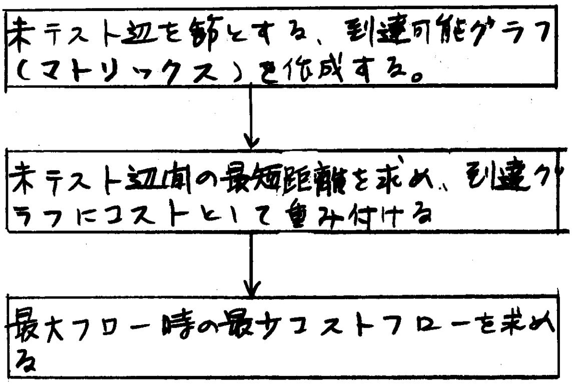
2のケースは、パス集合は2本となって最少のパス集合を導出しない。
3のケースは、テストパスに重複部が生じる。
このため、ケース2が生じないように自分自身には到達が出来ないように2部グラフを変形することと、又、ケース3が生じないようにするため，2部グラフの各辺に到達するための最短距離をコストとして重み付け、SからTへの最少コストフロー（フローは最大フロー）を流すことにより，最大許容セルを求める方法を導入する必要がある。 図－6はケース1では総コストが5となり、ケース2では総コストが8となるため、最少コストフローを流すことにより，ケース2は生じないことを示している。



図－6

6. 未テスト辺を含む最少で最短なS－Tパス集合の導出（最大許容せん法）

(1)、グラフに有向サイクルが含まれない場合



図－7

(2)、有向サイクルが含まれるのを許したグラフの場合

(1)に於いて 未テスト辺間の到達可能性が無いとして、(1)と同様に行う。

（本研究の未テスト辺のテスト法に関するもの以外は、文献（柳沢隆夫，パスカバー法によるプログラムテストデータの自動生成法に関して，芝浦工大研究報告理工系編，30巻，オ2号，1986）の1部である）

7、参考文献

[1]、R. K. Deb, On Generation of Test Data and Minimal Cover of Directed Graph, IFIP Congress Proceeding, pp. 13-16, 1977

[2] S. C. Ntatos and S. Louis Hakimi, On Path Cover Problems in Digraphs and Applications to Program Testing IEEE Transaction On Software Engineering, Vol. SE-5, NO5, pp520-529, 1979.

[3] 柳沢隆夫，プログラムテストに用いるパスジェネレータへのグラフ理論の応用（オ1報），芝浦工大研究報告理工系編，オ30巻，オ1号，1986.

[4] R. E. Prather, Theory of Program Testing-An Overview, THE BELL SYSTEM TECHNICAL JOURNAL, Vol. 62, No. 10, Part2, pp. 3073-3105, 1983.